PROJECT THEMIS: INFORMATION PROCESSING

AUBURN UNIVERSITY PROJECT THEMIS
TECHNICAL REPORT NUMBER AU-T-6

POLYNOMIAL MANIPULATION SYSTEM - FORTRAN IV PROGRAM

PREPARED BY

THE COMPUTER CENTER

LELAND H. WILLIAMS, DIRECTOR

AUBURN UNIVERSITY, AUBURN, ALABAMA

NOVEMBER, 1969

CONTRACT DAAH01-68-C-0296
ARMY MISSILE COMMAND
HUNTSVILLE, ALABAMA

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED.

APPROVED BY:

Leland H. Williams

L. H. Williams
Director, Computer Center
Associate Professor of
Mathematics
THEMIS Project Director

SUBMITTED BY:

Leland H. Williams

L. H. Williams
Director, Computer Center
Associate Professor of
Mathematics

Ben T. Lanham Jr

Ben T. Lanham, Jr.
Vice President for Research
Auburn University

AUBURN UNIVERSITY
AUBURN, ALABAMA 36830

FOREWORD

This report is a technical summary reporting the progress of a study conducted in the Mathematics Department and the Computer Center of Auburn University. The study is focused toward fulfillment of Contract No. DAAH01-68-C-0296 granted to Auburn University by the Army Missile Command, Huntsville, Alabama.

ACKNOWLEDGEMENT

The programming and documentation for this report were done by Mr. J. R. Sidbury.

ABSTRACT

A FORTRAN IV program which implements the Polynomial Manipulation System (PMS) is presented and described. PMS uses the Euclidean Algorithm to reduce a system of polynomials in several variables to a resultant system which can be solved sequentially as polynomials in one variable (Kronecker's method). PMS is described briefly and references are given to more complete discussions and to other pertinent literature.

TABLE OF CONTENTS

I. INTRODUCTION

The Polynomial Manipulation System (PMS) uses the Euclidean Algorithm for finding the eliminant and the greatest common divisor (g.c.d.) of two multi-variable polynomials. All polynomials involved are represented symbolically; PMS is a computer program whose input is the symbolic representation of two polynomials and whose output is (normally) the symbolic representation of their g.c.d. and eliminant.

The underlying theory and the application of PMS to the problem of solving systems of polynomial equations is discussed in [1], [2], and [3]. The present report describes a FORTRAN IV implementation of PMS developed on the IBM 360 Model 50 at Auburn University.

The program is described in Section II, the basic flow charts are given in Section III, Input/Output is discussed in Section IV, and efficiency of the method is discussed in Section V along with possible future work. The FORTRAN program is reproduced in Appendix A. Appendix B contains a simple example of the use of PMS for reduction of three polynomial equations in three variables to a resultant system which can be solved in sequence as polynomials in one variable.

II. PROGRAM DESCRIPTION

The PMS program is basically a main program with four subroutines, only one of which is significant. The other three subroutines are used for output, format headings on printed output and scaling of coefficients when they become large enough to possibly cause an overflow. In its present form the program is limited in that it is set up to use only 175K of IBM 360 storage. This limitation places constraints on the program which allows storage of only 50,000 polynomial entries (each term has n + 1 entries where n is the number of variables in the polynomial) which are presently set up as follows:

1) The pair of polynomials has, at most, four variables.

2) Each polynomial has at most 3160 terms.

3) The leading coefficient polynomials, to be defined below, can have, at most, 400 terms.

Minor modifications could increase the number of terms or variables or size of leading coefficients at the cost of decreasing the others or by use of a greater amount of machine storage. Still larger polynomials could be processed by use of tape, disc or other storage, but this has not been effected since such increases would only tend to accentuate certain disadvantages of the method to be discussed in Section V.

Consider the pair of polynomials $U,T : E^N \to R$. Let $x_1,\ldots,x_n$ denote the variables. Each of these polynomial functions can be considered as a polynomial in x_1 whose coefficients

would then be polynomial functions from E^{N-1} to R. Let U^0 and T^0 denote the polynomials in $x_2,\ldots,x_n$ which are the leading coefficients of U and T respectively considered as polynomials in x_1, and let u and t denote the degrees of U and T in x_1. We may assume $t \geq u$. Consider the polynomial R defined by

$$R = U^0T - T^0Ux_1^{t-u} \quad .$$

R is a polynomial in $x_1,\ldots,x_n$. Considering R as a polynomial in x_1 with polynomial coefficients, it is seen that Degree(R)<t. Let Degree(R) = r. If $r \geq u$, let T = R and repeat above procedure. If r<u, let T = U and U = R and repeat the above procedure. After a finite number of applications of this algorithm a polynomial R will be found whose degree in x_1 is zero. Thus R will be a polynomial in $x_2,x_3,\ldots,x_n$. It is easily seen that at each stage R has any zeros that are common to U and T. The R which is free of x_1 is called the eliminant of U and T.

III. BASIC FLOW CHARTS

The flowcharts for output and scaling will be omitted as their detail is not significant to the main purpose of the program. The main program flow chart is given on page 4.

MAIN PROGRAM

Read U,T

Compute U^0, T^0

Print U,T

1

Determine which of U and T has greatest degree in x_1. If it is T continue, if not interchange U and T, U^0 and T^0.

Call RESIDUE and Form R

Scale if necessary

If R is free of x_1, print R and return to beginning to read two new polynomials. If not, let T = R.

1

RESIDUE SUBROUTINE

Form U^0T

$\downarrow$

Form $-T^0Ux_1^{t-u}$

$\downarrow$

$R = U^0T - T^0Ux_1^{t-u}$

$\downarrow$

$T = R$

$\downarrow$

Calculate new T^0

$\downarrow$

Is $t = 0$?

no $\swarrow$ Return to Continue Processing

yes $\searrow$ Return. We Have Eliminant

IV. INPUT/OUTPUT

The polynomials U and T are read in separately. The first card for each polynomial contains the number of variables in the polynomial in columns 31-34 in integer format, right justified. The number of terms in the polynomial appear in columns 35-38 in integer format, right justified. Following this card the terms of the polynomial appear, one term per card. The coefficient appears in columns 1-16 in E format; following this are the exponents of the variables right justified in integer format in columns 17-21, 22-26, 27-31, etc.

The output of this program is available in any medium, although the program is currently set up for printed output only.

V. EFFICIENCY AND FUTURE WORK

The PMS program has not proved useful as a method of reducing polynomial systems of equations to a resultant system for the following reasons:

1) Storage efficiency is low. An inordinate amount of core is needed to process many simple appearing systems of equations.

2) Time efficiency is low. Extreme amounts of time are needed to solve all but the most simple problems. As few as four equations in four variables with small exponents (on the order of ten or less) take many hours of machine time to reach a solution. Simpler problems are solvable in

small amounts of time, but other methods without these disadvantages can be used to solve these systems.

3) Certain types of systems give solutions which have a low order of accuracy. Several articles, [4], [5], [6], have been published discussing this problem as well as the two above.

The basic PMS program will be examined and modified to determine if it is of value in algebraically solving simple systems of differential equations.

VI. REFERENCES

1. Williams, L. H. Algebra of Polynomials in Several Variables for a Digital Computer, Duke University Memorandum 61-5-5,1 (1 May 1961).
2. Williams, L. H. Algebra of Polynomials in Several Variables for a Digital Computer, J. ACM 9 (1962), 29-40.
3. Williams, L. H. Non-Numeric Mathematical Computation, Auburn University Report No. AU-T-2, The First Annual Report on Project THEMIS Research at Auburn University (October 31, 1968), 175-185.
4. Emmer, Edward J., Jr. Multiple Solution Sets for Non-linear Mechanics Problems, Ph.D. Thesis, Case Institute of Technology, 1966.
5. Collins, George T. Subresultants and Reduced Polynomial Remainder Sequences, J. ACM (Jan. 1967), 128-142.
6. Ku, S. Y. and Adler, R. J. Computing Polynomial Resultants: Bezout's Determinant vs. Collins' Reduced P.R.S. Algorithm, Comm. ACM (Jan. 1969), 23-30.

APPENDIX A

FORTRAN IV Program Listing

```
C***********************************************************************
C***********************************************************************
C
      IMPLICIT INT[illegible](I-N)
      COMMON UMA[illegible],I[illegible],ITMAX(2,400),IM[illegible](400),R(3160),
     1IR(4,3160),ITRAT[illegible],I[illegible],[illegible](3160),I[illegible](4,3160),[illegible],IT(4,3160),
     1LS2,LS1,LM
      MAXP = 400
C***********************************************************************
C***********************************************************************

1[illegible]   I[illegible] = [illegible]
      LM=0
      I[illegible] = [illegible]
[illegible]   READ(5,1) [illegible]
      IF(NVAR1.[illegible]) [illegible]
     LS1=0
     LS2=[illegible]
      DO[illegible]I=1,[illegible]
[illegible]   READ(5,4) [illegible](I),(I[illegible](I,J),J=1,NVAR1)
      READ(5,1) [illegible]
      DO[illegible]I=1,[illegible]
[illegible]   READ(5,4) [illegible](I),(I[illegible](I,J),J=1,NVAR2)
      DO 7 J = 1,NT[illegible]
      R(J)=U(J)
      DO7I=1,NVAR1
7     IR(I,J)=I[illegible](I,J)
      IF(NVAR1 .GE. NVAR2) GO TO 106
      MAX = NVAR2
      NVAR1 = NVAR1 + 1
      DO 10[illegible] J = NVAR1 ,MAX
      DO 10[illegible] K = 1,[illegible]
[illegible]   IR(J,K) = 0
      NVAR1=MAX
106   MAX = NVAR1
      JPWRU = 0
      DO 50 J =1,NTERM1
      IF(JPWRU - IR(1,J)) 49,50,50
49    JPWRU = IR(1,J)
50    CONTINUE
      JU = 0
      DO 55 J = 1, NTERM1
      IF(JPWRU - IR(1,J)) 55,54,55
54    JU = JU + 1
      UMAX(JU) = R(J)
      IF(ABS(UMAX(JU)).GE.1.E6) LM=1
      DO 53 K = 2,MAX
```

```
53      IUMAX(K-1,JU) = IR(K,J)
55      CONTINUE
        DO 10 J = 1,NTERM1
        U(J)=R(J)
        DO10I=1,MAX
10      IU(I,J)=IR(I,J)
        DO  9 J = 1,NTERM2
        R(J)=T(J)
        DO9I=1,NVAR2
9       IR(I,J)=IT(I,J)
        CALL PRINT(2,NTERM2,0,0,NVAR2)
        IF(NVAR2 .GE. NVAR1)  GO TO 1075
        MAX = NVAR1
        NVAR2 = NVAR2 + 1
        DO 104 J = NVAR2 ,MAX
        DO 104 K = 1,NTERM2
104     IR(J,K) = 0
        NVAR2 = MAX
1075    MAX = NVAR2
        JPWRT = 0
        DO 60 J =1,NTERM2
        IF(JPWRT - IR(1,J)) 51,60,60
51      JPWRT = IR(1,J)
60      CONTINUE
        JT = 0
        DO 56 J = 1, NTERM2
        IF(JPWRT - IR(1,J)) 56,57,56
57      JT = JT + 1
        TMAX(JT) = R(J)
        IF(ABS(TMAX(JT)).GE.1.E6) LM=1
        DO 58 K = 2,MAX
58      ITMAX(K-1,JT) = IR(K,J)
56      CONTINUE
        DO  8 J = 1,NTERM2
        T(J)=R(J)
        DO8I=1,MAX
8       IT(I,J)=IR(I,J)
107     IF(JPWRT-JPWRU) 70,71,71
70      NN = JPWRT
        JPWRT = JPWRU
        JPWRU = NN
        MAXT=NTERM1
        IF(NTERM2.GT.NTERM1) MAXT=NTERM2
        DO80I=1,MAX
        TEMP=U(I)
        U(I)=T(I)
        T(I)=TEMP
        DO80J=1,MAXT
        TEMP=IU(J,I)
        IU(J,I)=IT(J,I)
80      IT(J,I)=TEMP
        NN = JU
        JU = JT
        JT = NN
        NN = NTERM1
        NTERM1 = NTERM2
        NTERM2 = NN
        NN = NVAR1
```

```
      NVAR1 = NVAR2
      NVAR2 = NN
      JJ=JU
      IF(JT.GT.JJ) JJ=JT
      MA = MAX-1
      DO 73 J = 1, JJ
      TW=UMAX(J)
      UMAX(J) = TMAX(J)
      TMAX(J)=TW
      DO73K=1,MA
      ITT=IUMAX(K,J)
      IUMAX(K,J) = ITMAX(K,J)
73    ITMAX(K,J)=ITT
71    CONTINUE
C
      IF(LM.EQ.1) CALL SCALE2(NTERM1,NTERM2,MAX)
      LM=0
  101 CALL RESIDU (NTERM1,NTERM2,MAX,JPWRU,JPWRT,JU,JT)
      IF(LM .EQ. 1) CALL SCALE2(NTERM1,NTERM2,MAX)
      IF(LM.EQ.1) GOTO 3001
      IF(LS2.EQ.1) CALL SCALE2(NTERM1,NTERM2,MAX)
      LS2=0
3001  LM=0
300   IF(LS1.EQ.0) GOTO 107
      LS1=0
      LM=0
  301 CALL PRINT ( 3,JU,NTERM1,NTERM2,MAX)
      WRITE(6,10000) ITIMES
10000 FORMAT(1H , 5HSCALE, I3)
      GO TO 100
    1 FORMAT(30X,2I4)
4     FORMAT(E16.7,10I5)
      END
```

```
C         SUBROUTINE FOR PRINTING THE TWO POLYNOMIALS AND THE RESIDUE
C
C
          SUBROUTINE PRINT (L,JU,NTERM1,NTERM2,N)
C***********************************************************************
C***********************************************************************
C
          IMPLICIT INTEGER*2(I-N)
          COMMON IUMAX(400),IUMAX(3,400),ITMAX(3,400),TMAX(400),R(3160),
     NIR(4,3160),ITRATE,ITIMES,U(3160),IU(4,3160),T(3160),IT(4,3160),
     NLS2,LS1,L4
C
C***********************************************************************
C***********************************************************************
C
          IF(L.EQ.3) WRITE(6,310)
          IF(L.EQ.2) WRITE(6,311)
          IF(L.EQ.1) WRITE(6,312)
   50 CALL PRINTH(N)
          DO315K=1,JU
          WRITE(6,43 ) R(K),(IR(I,K),I=1,N)
315       CONTINUE
        K = N-1
503       CONTINUE
          RETURN
   43 FORMAT(1X,E16.7,3X,10(I5,5X))
311       FORMAT(1H1,'THE POLYNOMIAL   U   IS')
312       FORMAT(1H-,'THE POLYNOMIAL   T   IS')
310       FORMAT(1H-, 'THE ELIMINANT IS')
  540 FORMAT (20X,2I4)
  541 FORMAT(E16.7,10I5)
        END
```

```
C       SUBROUTINE FOR PRINTING HEADINGS
      SUBROUTINE PRINTH(K)
C
C                   THIS SUBROUTINE MERELY PRINTS COLUMN HEADINGS FOR THE
C                   VARIABLES DEPENDING ON THE NUMBER OF VARIABLES.
C                   THAT IS ITS ONLY PURPOSE.  IT WILL HANDLE UP TO 10
C                   VARIABLES.
C
      GO TO (21,22,23,24,25,26,27,28,29,30),K
21     WRITE(6,31)
      RETURN
22     WRITE(6,32)
      RETURN
23     WRITE(6,33)
      RETURN
24     WRITE(6,34)
      RETURN
25     WRITE(6,35)
      RETURN
26     WRITE(6,36)
      RETURN
27     WRITE(6,37)
      RETURN
28     WRITE(6,38)
      RETURN
29     WRITE(6,39)
      RETURN
30     WRITE(6,40)
      RETURN
   31 FORMAT(1H0,11HCOEFFICIENT,10X,4HX(1))
   32 FORMAT(1H0,11HCOEFFICIENT,10X,4HX(1),5X,4HX(2))
   33 FORMAT(1H0,11HCOEFFICIENT,10X,4HX(1),5X,4HX(2),5X,4HX(3))
   34 FORMAT(1H0,11HCOEFFICIENT,10X,4HX(1),5X,4HX(2),5X,4HX(3),5X,4HX(4)
     1)
   35 FORMAT(1H0,11HCOEFFICIENT,10X,4HX(1),5X,4HX(2),5X,4HX(3),5X,4HX(4)
     1,5X,4HX(5))
   36 FORMAT(1H0,11HCOEFFICIENT,10X,4HX(1),5X,4HX(2),5X,4HX(3),5X,4HX(4)
     1,5X,4HX(5),5X,4HX(6))
   37 FORMAT(1H0,11HCOEFFICIENT,10X,4HX(1),5X,4HX(2),5X,4HX(3),5X,4HX(4)
     1,5X,4HX(5),5X,4HX(6),5X,4HX(7))
   38 FORMAT(1H0,11HCOEFFICIENT,10X,4HX(1),5X,4HX(2),5X,4HX(3),5X,4HX(4)
     1,5X,4HX(5),5X,4HX(6),5X,4HX(7),5X,4HX(8))
   39 FORMAT(1H0,11HCOEFFICIENT,10X,4HX(1),5X,4HX(2),5X,4HX(3),5X,4HX(4)
     1,5X,4HX(5),5X,4HX(6),5X,4HX(7),5X,4HX(8),5X,4HX(9))
   40 FORMAT(1H0,11HCOEFFICIENT,10X,4HX(1),5X,4HX(2),5X,4HX(3),5X,4HX(4)
     1,5X,4HX(5),5X,4HX(6),5X,4HX(7),5X,4HX(8),5X,4HX(9),5X,5HX(10))
      END
```

```
C         RESIDUE SUBROUTINE      FORMS RESIDUE AND SORTS TERMS
C
      SUBROUTINE RESIDU(NTU,NTT,NV,JPU,JPT,JU,JT)
C*********************************************************************
C*********************************************************************
C
      IMPLICIT INTEGER*2(I-N)
      COMMON UMAX(400),IUMAX(3,400),ITMAX(3,400),TMAX(400),R(3160),
     NIR(4,3160),ITRATE,ITIMES,U(3160),IU(4,3160),T(3160),IT(4,3160),
     NLS2,LS1,LM
      DIMENSION  IB(5)
      MAXR=3160
      MAXP = 400
      INTEGER  SWITCH
C
C*********************************************************************
C*********************************************************************
C
      I =1
      DO 20 J = 1,NTT
      A=T(J)
      DO5210 K=1,NV
5210  IB(K)=IT(K,J)
      IF(IB(1).GE.JPT)GOTO20
      IF(ABS(A) - 1.E6)  45,46,46
46    LS2=1
      RETURN
45    DO 19 L = 1, JU
      R(I) = UMAX(L)*A
      IR(1,I) = IB(1)
      DO 10 KK = 2,NV
10    IR(KK,I) = IUMAX(KK-1,L) + IB(KK)
      MM=I-1
      IF(MM.EQ.0)GOTO19
      DO80KK=1,MM
      IF(IR(1,KK).NE.IR(1,I))GOTO80
      DO700KKK=2,NV
      IF(IR(KKK,KK).NE.IR(KKK,I))GOTO80
700   CONTINUE
      R(KK)=R(KK)+R(I)
      IF(ABS(R(KK)).GT.1.E-9)GOTO702
      NNM=MM-1
      DO800ILK=KK,NNM
      R(ILK)=R(ILK+1)
      DO800NPQ=1,NV
800   IR(NPQ,ILK)=IR(NPQ,ILK+1)
      I=I-1
702   I=I-1
      GOTO701
80    CONTINUE
701   IF(I.LT.MAXR)GOTO19
11    WRITE(6,12)
12    FORMAT(1H1,25HTOO MANY TERMS IN RESIDUE)
      STOP
19    I = I + 1
20    CONTINUE
      DO 40 J = 1,NTU
      A=U(J)
```

```
      DO5211 K=1,NV
5211  IB(K)=IP(K,J)
      IF(IB(1).GE.JPU)GOTO40
      IF(ABS(A) - 1.E6) 47,48,48
48    LS2=1
      RETURN
47     DO  41  L = 1, JT
      R(I) = -TMAX(L)*A
      IR(1,I) = IB(1) + JPT -JPU
      DO 30  K = 2,NV
30    IR(K,I) = ITMAX(K-1,L) + IB(K)
      MM=I-1
      DO380KK=1,MM
      IF(IR(1,KK).NE.IR(1,I))GOTO380
      DO3700KKK=2,NV
      IF(IR(KKK,KK).NE.IR(KKK,I))GOTO380
3700  CONTINUE
      R(KK)=R(KK)+R(I)
      IF(ABS(R(KK)).GT.1.E-9)GOTO3702
      NNM=MM-1
      DO3800ILK=KK,NNM
      R(ILK)=R(ILK+1)
      DO3800NPO=1,NV
3800  IR(NPO,ILK)=IR(NPO,ILK+1)
      I=I-1
3702  I=I-1
      GOTO3701
380   CONTINUE
3701  IF(I.LT.MAXE)GOTO41
      GOTO11
41    I = I + 1
40    CONTINUE
      I = I - 1
99    JPWRT=0
      JK=0
100   DO 105  M = 1,I
      IF(IR(1,M) - JPWRT)  105,105,102
102   JPWRT = IR(1,M)
105   CONTINUE
      JPT = JPWRT
      IF(JPJPT)130,142,130
142   LS1=1
      JU = I
      RETURN
130   JT = 0
      DO7II=1,400
      TMAX(II)=0.
      DO7JJ=1,3
7     ITMAX(JJ,II)=0
25    DO56J=1,I
      IF(JPWRT - IR(1,J)) 56,57,56
57    JT = JT + 1
      TMAX(JT) = R(J)
      IF(TMAX(JT).GE.1.E6) LM=1
      DO58 K=2,NV
58    ITMAX(K-1,JT) = IR(K,J)
      CONTINUE
56    CONTINUE
```

```
      DO1J=1,I
      T(J)=R(J)
      DO1L=1,NV
1     IT(L,J)=IR(L,J)
376   NTT=I
      RETURN
      END
```

```
      SUBROUTINE   SCALE2(NTERM1,NTERM2,NV)
      IMPLICIT INTEGER*2(I-N)
      COMMON UMAX(400),IUMAX(3,400),ITMAX(3,400),TMAX(400),R(3160),
     NIR(4,3160),ITRATE,ITIMES,U(3160),IU(4,3160),T(3160),IT(4,3160),
     NLS2,LS1,LM
      EQUIVALENCE(NVAR1,MAX)
      MAX = NV
      NVAR2=NVAR1
      DO 1 I = 1,400
      UMAX(I) = UMAX(I)/1000.
1     TMAX(I) = TMAX(I)/1000.
      DO  7 J = 1,NTERM1
7     U(J) = U(J)/1000.
      DO  9 J = 1,NTERM2
9     T(J) = T(J)/1000.
      ITIMES = ITIMES + 1
      RETURN
      END
```

APPENDIX B

This appendix presents an example problem. The following system of three polynomials in three variables is reduced to a single polynomial in one variable:

$$(1)\quad x_1 + x_2 + x_3 = 0$$

$$(2)\quad x_1 + x_2^2 = 0$$

$$(3)\quad x_1 + x_3^2 = 0$$

First, x_1 is eliminated between (1) and (2) producing the following printout which has been labeled for expository convenience:

COEFFICIENT		x(1)	x(2)	x(3)	
0.1000000E	01	1	0	0	(1)
0.1000000E	01	0	1	0	
0.1000000E	01	0	0	1	

COEFFICIENT		x(1)	x(2)	x(3)	
0.1000000E	01	1	0	0	(2)
0.1000000E	01	0	2	0	

COEFFICIENT		x(1)	x(2)	x(3)	
0.1000000E	01	0	2	0	(E_1)
-0.1000000E	01	0	1	0	
-0.1000000E	01	0	0	1	

Second, x_1 is eliminated between (3) and (2) producing the following printout:

COEFFICIENT		x(1)	x(2)	x(3)	
0.1000000E	01	1	0	0	(3)
0.1000000E	01	0	0	2	

COEFFICIENT		x(1)	x(2)	x(3)	
0.1000000E	01	1	0	0	(2)
0.1000000E	01	0	2	0	

COEFFICIENT		x(1)	x(2)	x(3)	
0.1000000E	01	0	2	0	(E_2)
-0.1000000E	01	0	0	2	

Finally (E_1) and (E_2) are treated as a pair of polynomials in two variables x_1 and x_2. Then x_1 (x_2 in our first system) is eliminated, producing the following printout:

COEFFICIENT		x(1)	x(2)	
0.1000000E	01	2	0	
-0.1000000E	01	1	0	(E_1)
-0.1000000E	01	0	1	

COEFFICIENT		x(1)	x(2)	
-0.1000000E	01	2	0	(E_2)
0.1000000E	01	0	2	

COEFFICIENT		x(1)	x(2)	
0.1000000E	01	0	4	
-0.2000000E	01	0	3	(E_3)

(E_3) is our eliminant free of x_1 and x_2, so it can be solved. Using its solution (E_2) can then be solved. Using this (2) can be solved and the solutions to the resultant system (2), (E_2), (E_3) are the solutions to the system (1), (2), (3). Three other equations from these six could have been taken to form the resultant system.

UNCLASSIFIED
Security Classification

APPENDIX C

DOCUMENT CONTROL DATA - R & D (Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)		
1. ORIGINATING ACTIVITY (Corporate author) Auburn University Auburn, Alabama	2a. REPORT SECURITY CLASSIFICATION Unclassified	
	2b. GROUP N/A	
3. REPORT TITLE POLYNOMIAL MANIPULATION SYSTEM-FORTRAN IV PROGRAM		
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Technical Report		
5. AUTHOR(S) (First name, middle initial, last name) Leland H. Williams, James R. Sidbury		
6. REPORT DATE November 1969	7a. TOTAL NO. OF PAGES 23	7b. NO. OF REFS 6
8a. CONTRACT OR GRANT NO. DAAH01-68-C-0296	9a. ORIGINATOR'S REPORT NUMBER(S) AU-T-6	
b. PROJECT NO. N/A		
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) None	
d.		
10. DISTRIBUTION STATEMENT Distribution of the document is unlimited.		
11. SUPPLEMENTARY NOTES None	12. SPONSORING MILITARY ACTIVITY Army Missile Command	

13. ABSTRACT

A FORTRAN IV program which implements the Polynomial Manipulation System (PMS) is presented and described. PMS uses the Euclidean Algorithm to reduce a system of polynomials in several variables to a resultant system which can be solved sequentially as polynomials in one variable (Kronecker's method). PMS is described briefly and references are given to more complete discussions and to other pertinent literature.

DD FORM 1 NOV 65 1473 REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

UNCLASSIFIED
Security Classification

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Polynomials						
Resultant						
Eliminant						
Euclid's Algorithm						
FORTRAN						
Program						

UNCLASSIFIED

Security Classification